内网 IDS 蜜罐型集成系统

◆孙韬

（成都东软学院 网络空间研究所）

摘要：为了应对当前日益严峻的内网安全问题，我们制作了内网蜜罐集成系统，该集成系统应用第三代蜜网部署为基础，配合 IDS 和防火墙来构建，使用蜜网网关转移攻击流量，建立多台蜜罐服务器，能够第一时间检测并警告管理员，使应急响应满足 P2DR 模型的防护时间大于检测加响应时间，分析并记录攻击手段，实现了证据留存安全迅速和内网其他主机的安全。

0 引言

集成蜜罐系统（Honeypot system integration）作为被动 IDS 和引诱攻击目标，在技术日趋成熟的同时也面临着安全方面的挑战，蜜罐系统不但会收到常规手段的入侵，也会受到大量未知的攻击手段，现在大量采用的防御方案是是传统的（IDS Inrusion Detection Systems）和防火墙，以及一些支持安全框架服务。这些防御方式都是对外来攻击进行处理，并不会从攻击手段来进行分析，企业内网安全也是防火墙所不能接手的，根据短板效应，如若内网一台机器沦陷后，内网的其他服务器及 PC 都有被渗透的危险。对此，本论文提出蜜罐系统与防火墙结合对内网安全事件进行记录和应急响应，通过虚拟化技术将制作成本大幅降低，在企业内部机房下对整体网络影响也降到最低，使用反向链接 waf 对将来扩展大量 web 服务留下空间，构建内网集成蜜罐系统来防御域（组）渗透。

1 蜜罐系统

蜜罐系统是在网络安全发展过程中催生出的一种技术，通过一些已知的漏洞或者常见服务来引用黑客进行攻击，当攻击者入侵后，你可以通过日志和蜜罐的管理端对黑客入侵的手段和 0day 进行 payload 分析，从而进行相关防护。蜜罐系统如今已经发展到第三代蜜罐，蜜罐技术分为低交互蜜罐和高交互蜜罐，低交互蜜罐的意思是通过模拟这个服务的端口和返回数据包来达到迷惑入侵者的目的，引诱入侵者来进行溢出攻击等。高交互蜜罐则是一个真实的服务，黑客可以对其进行入侵，由于是蜜罐系统。当黑客进行深度操作后系统会自动断开连接，并提醒管理员。蜜罐系统的功能在于迷惑入侵者，保护服务器，和增加入侵者的入侵时间提高应急响应反应时间，加固服务器。

1.1 蜜罐技术

蜜罐技术是一种新型并且技术成熟的网络安全主动防御技术。蜜罐技术通过构造常见漏洞特征码来做到引诱攻击者攻击自身，捕获攻击数据、攻击方式、攻击目的等攻击方面的相关信息。蜜罐系统被探测、攻击、入侵的时候，他的作用才会体现出。

蜜罐系统具有以下特征：

其中重要的一点就是蜜罐系统是存在漏洞，完整的暴露在内网或外网中的机器

蜜罐系统的机器是虚假的，攻击者需要花费时间攻破，通过 P2DR 模型可知，当攻击时间大于响应时间时，系统是安全的。在攻击的这段时间内，系统管理员能够有足够的时间锁定攻击来源，并保护内网其他生产主机。

能够学习攻击者的攻击方法和利用手法。

利用数据捕获可以捕获攻击者使用的 0day，通过分析数据内容来对位置漏洞进行修复。[1]

1.2 蜜罐的关键技术

想要构建一个蜜罐系统和组成一个内网的蜜罐网络主要技术在于主机虚拟化，攻击数据控制，攻击数据捕获和危险操作隔离，这四个方面是组成内网蜜罐系统的主要技术以及主要功能。

虚拟化分为主机虚拟化和功能模拟，其目的是减少内网服务器资源浪费，通过虚拟化技术将蜜罐系统同步上线到相关服务器同网段，若上线服务器被入侵时，攻击者对内网进行渗透时蜜罐系统保证能够被入侵者发现。[2]功能模拟要做到能够有明显的漏洞指纹，通过同网段下其他服务器用 iptables 或 firewall 等防火墙设置，进行流量或者探测转发（nginx 反向代理），将入侵数据全部转移到蜜罐系统中。由于伪蜜罐系统存在用户交互问题，但是对蜜罐系统较为安全，所以本文大部分的蜜罐系统都

收稿日期：2017 年 8 月 25 日 孙韬 1997 年出生 性别男 学生 主要研究网络空间安全攻防模拟 admin@8sec.ac.cn

是中，低交互的蜜罐系统，防止攻击者攻击得到蜜罐系统权限。

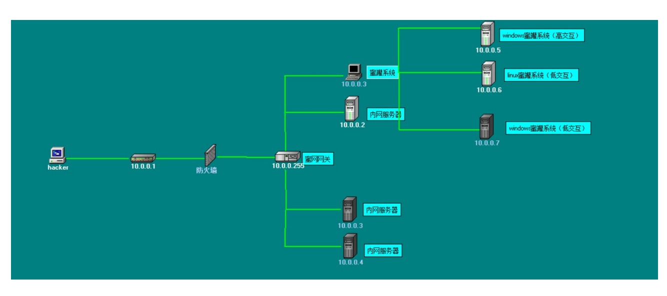
攻击数据控制的目的是保证蜜罐系统被攻击时不会被攻击者用作跳板来攻击内网的其他业务服务器。这样就决定了数据控制需要有两个以上的控制方案同时进行，比如从网关防火墙处进行流量控制，同时从内网服务器之间的软防火墙（WAF）对数据传输设置 IP 白名单。

攻击数据捕获是蜜罐系统的主要功能，捕获的数据要及时发送到管理员邮箱中，使管理员知晓内网已经被攻击了，执行应急响应流程。捕获的数据用来分析攻击者的攻击手法、攻击工具和攻击目等其他目的。通过链路层到应用层，整个内网系统记录攻击者的数据内容，多维度数据捕获模式，保证攻击者的每一个攻击都有记录。

攻击者通过蜜罐系统的漏洞拿到了蜜罐系统的权限，当攻击者需要做跳板攻击其他服务器等危险操作时，蜜罐系统会自动断开与攻击者的连接，这样保证了蜜罐系统和内网其他服务器的安全，保证危险操作能被隔离出来。

2 系统以及结构设计

内网系统的集成蜜罐的设计如图 1-1 所示，



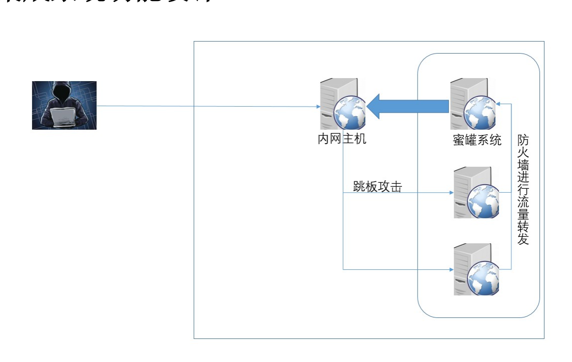
1- 0

系统主要由防火墙，蜜网网关，蜜罐系统管理端，虚拟化三台高低交互蜜罐系统组成，防火墙由硬件防火墙充当，用于记录传输层攻击命令，并进行流量转发，加强网络之间的访问控制，预防系统外部网络入侵内部网络或非法操作单机的一种必要防护。防火墙对所有流经数据都进行记录。

入侵检测技术是一种通过记录攻击者相关信息观察攻击行为，检测鉴定是否为攻击行为的一种检测技术，而蜜罐系统则是对所有信息进行记录，通过设定的正则规则判断是否是攻击行为，内网任意一台服务器如果被外部攻击者攻击，开始对内网服务器进行扫描和溢出等操作时，通过控制的蜜网网关和服务器部署的分布式 WAF 将非法流量转移到蜜罐系统中，确保攻击者的所有攻击目标全部都转移到蜜罐系统上。[3]

蜜罐系统控制端（10.0.0.3）主要是虚拟化生产蜜罐系统，并实时监测记录，所收集的数据大多是扫描记录，相关危险端口登陆记录，登陆后命令操作，溢出数据包内容。蜜网网关也可通过虚拟化技术或者使用系统内部集成来部署，例如 IPTables 防火墙，firewall，winshark 内网数据监测分析工具。通过这些工具来对蜜网数据进行监控。

3 集成系统功能设计



假设攻击者在内网已经获得一台服务器的 root 权限，且与集成蜜罐系统在同一网段下，攻击者的攻击手段大多是依靠服务器上各种服务端口漏洞或者弱口令进行攻击，所以我们在每个蜜罐主机上部署了一些可用的漏洞，提高了蜜罐的诱惑性，攻击者会对蜜罐系统的网络及新端口扫描，获取蜜罐系统的端口信息，当系统接收到 syn 或者 ping 命令探测后立即对管理员进行告警。WAF 使用 nginx 反向连接的模式，并没有采用通明网桥或者离线模式，如果采取离线模式，那么只是对攻击起到记录作用，而没有起到阻隔危险代码的作用并且后期维护工作量大。如采取通明网桥模式则后期升级成本过高，若防火墙流量有限制则对大量并发连接处理不够，而采取反向连接式 waf 则能够部署分布式集群处理大量连接用户，未来升级或者访问量过大时只需添加新的 waf 服务器即可，openresty 配合 nginx 可以轻松处理 100000+的并发连接数，对后期升级提供了简便并低成本的解决方案。

Web 防火墙（WAF）主要对 web 特有的入侵方式进行防护，如 cc 攻击，sql 注入，xml 注入，xss 注入等。由于是应用层攻击，所以也可以称作 web IPS, WAF 从技术特点分为以下几部分：

代理服务：基于双向代理，中断了用户与服务器的直接连接，从而防止了入侵直接作用于 web 服务器，分布式的 waf 对 cc 攻击也有一定的阻隔效果。

特征识别：识别出入侵者是他防护的功能，特征就

是攻击者的“指纹”，如缓冲区溢出的 shellcode，sql 注入中的‘=’‘select’等。通过正则表达式判断是否为规则库中的内容来返回相应的内容。

识别库：拥有大量的特征识别库也是一个 WAF 所拥有的核心竞争之一。

WAF 部署位置决定 WAF 部署位置的是 WEB 服务器的位置。因为 WEB 服务器是 WAF 所保护的目标，部署时要尽量接近。

WAF 部署种类和原理

网桥代理模式：当客户端主机对服务器有连接请求时，tcp 连接请求被 WAF 截取和监控。WAF 可以看作一段网线，通过双网卡基于桥模式进行转发。从客户端主机来看，客户机仍然是直接访问服务器，察觉不到 WAF 的存在。

优点：无丢包问题，默认支持 https，可主动防御。

缺点：由于所有流量都将途径 WAF，而导致防火墙硬件处理能力将限制流量带宽，应对大规模并发连接时力不从心。

反向代理模式：该模式是指将真实服务器的地址映射到反向代理服务器上。此时客户主机访问的就是代理服务器，代理服务器再将数据包转发给后台服务器，后台服务器收到请求后将响应在发送给代理服务器，由代理服务器在传给客户端。

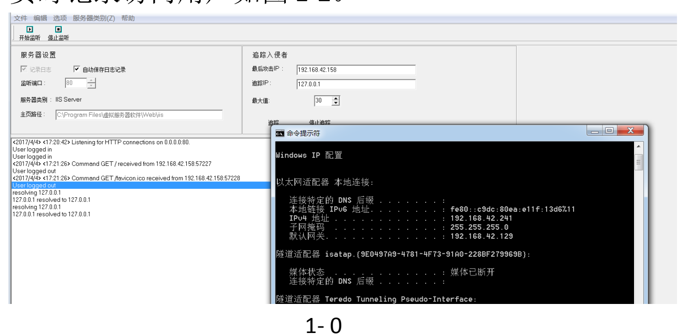
优点：能够隐藏后台服务器 ip，防止溢出漏洞直接作用于服务器。由于代理服务器可以组成集群，对并发连接进行负载均衡，升级成本降低。（也是由于能够隐藏后台主机 IP，所以本论文采用反向代理模组建 WAF）。

缺点：部署复杂，要在服务器上完成带路部署，设置代理端口和服务器端口。

4 实验结果与分析

主机名称	系统要求	环境要求	硬件要求
Linux 主机 1（linux 蜜罐）	centos6.5	kippo，dionaea，dinaeaFR	1G RAM 10G 硬盘 2*2CPU
Liunx 主机 2（WAF）	centos6.8	nginx、openresty、x-waf、mysql、php	1G RAM 10G 硬盘 2*2CPU
Windows 主机（内网上线主机）	Windows2003 server	Apache2.1+mysql+php	4G RAM 40G 硬盘 2*2CPU
Windows 主机（高交互蜜罐）	Windows2003 server	iis6.0+accsse+南方数据 cms，Server-U6.0 以下版本	1G RAM 20G 硬盘 2*2CPU
Windows7 (低交互蜜罐)	Windows7	Ts 虚拟服务器（模拟 iis7.5）	1G RAM 20G 硬盘 2*2CPU
Linux 主机 1（linux 蜜罐）	centos6.5	kippo，dionaea，dinaeaFR	1G RAM 10G 硬盘 2*2CPU

实时记录访问用户如图 1-2。



1- 0

Windows 7（模拟攻击者沦陷内网机器）：

1g 内存 20g 硬盘

安装 sqlmap，nc，nmap，winshark，burpsuit

本实验在一台主机上，利用 vmware 做虚拟化部署，首先主机安装 VMwawre 12。之后建立五台虚拟机，分别为 windows 蜜罐（低交互），linux 蜜罐（低交互），windows 蜜罐（高交互），windows2003（假设被沦陷内网主机），linux 云 WAF 主机。

Linux 蜜罐系统我选择 ubuntu IP: 192.168.42.63

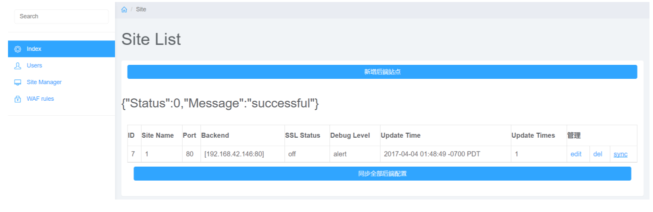
Linux WAF 选择 centos6.8 做管理端 IP:192.168.42.6

Windows2003 IP: 192.168.42.146

Windows 蜜罐 IP:192.168.42.241

在 waf 上配置后端服务器，使用 nginx 反向代理防止恶意攻击影响本地服务器。

WAF 采取了 openresty +集成 nginx+mysql 构成，采用 lua 和 go 语言写入规则。



1- 0

使用 WAF 后将正常上线的服务器添加到 waf 反向链接中，如图 1-3，设置 windows2003-防火墙禁止内网网段 ip 访问，并使用 403 将内容转发到 windwos 高交互蜜罐的网站中。

Linux 低交互蜜罐配置了 kippo 蜜罐和 dionaea，通常攻击者会对 ssh 进行爆破攻击，所以使用这个 ssh 蜜罐来应对这种情况。而 dionaea 来模拟各种常见端口和漏洞，并且使用 webgui 设计进行后端管理。开启 Kippo：

#cd /honeydrive/kippo

#./start.sh

开启 dionaea：

#cd /opt/Dionaea/bin

#sudo ./Dionaea –l all

开启 dionaeaFR

cd /opt/DionaeaFR/

python2.7 manage.py collectstatic

python2.7 manage.py runserver 0.0.0.0:8000

实验 1，在试验内网蜜罐系统的被攻击的主机对系统中的三台蜜罐系统使用 nmap 端口扫描

实验环境：windows7（攻击主机）

Nmap 端口扫描工具

IP：192.168.42.100

Ubuntu （低交互蜜罐系统）

Dionaea+mysql+python3.2/pthon2.7+DionaeaFR

IP：192.168.42.63

假设内网某台主机（IP：192.168.42.100）被攻击者拿到管理员权限，并下载了 nmap，准备对内网某台服务器（IP：192.168.42.63）进行端口扫描，则执行命令

#nmap –Ss 192.168.42.63

扫 描 结 果 如 图 1-4 。

```
C:\Windows\system32\cmd.exe
C:\Users\DELL\Desktop\nmap-6.46>nmap -sS 192.168.42.63

Starting Nmap 6.46 ( http://nmap.org ) at 2017-04-04 18:31 中国标准时间
Nmap scan report for 192.168.42.63
Host is up (0.00s latency).
Not shown: 988 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
42/tcp   open  nameserver
80/tcp   open  http
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
1433/tcp open  ms-sql-s
3306/tcp open  mysql
5060/tcp open  sip
5061/tcp open  sip-tls
8000/tcp open  http-alt
MAC Address: 00:0C:29:F7:BE:76 (VMware)

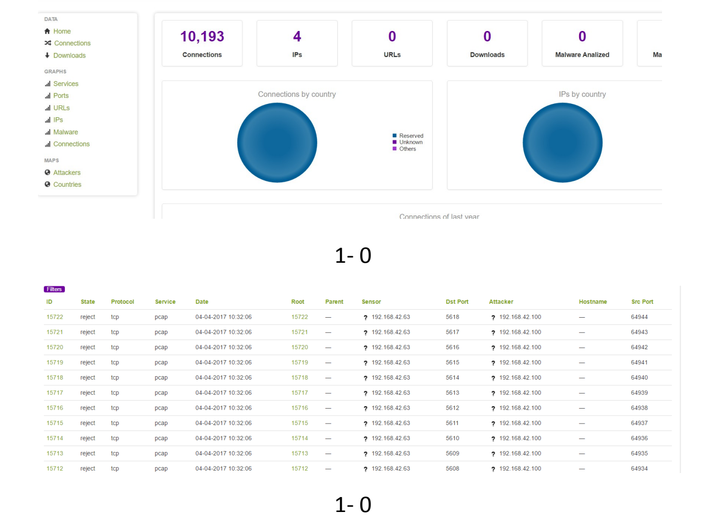
Nmap done: 1 IP address (1 host up) scanned in 13.49 seconds

C:\Users\DELL\Desktop\nmap-6.46>X_
```

1- 0

可以看出，蜜罐系统已经模拟出相关的服务信息了，这时攻击者对服务器的相关数据操作已经可以从后台查看到如图 1-5、图 1-6。

Nmap 的扫描已经被蜜罐系统记录，而且被迷惑模拟出这些服务端口。



1- 0

1- 0

实验 2，对 windows 内网服务器网站进行注入操作。

实验环境：linux 主机（cetos6.8 WAF 代理服务器）

Nginx+openresty+mysql5 端口配置 80，远程代理 80 端口

IP:192.168.42.6

Linux 服务器（windows 2003 server sp2）

Apache+mysql+php 正常生产主机。

漏洞源码（dedecms2007 php 版本）

IP：192.168.42.148

假设攻击者拿到内网某服务器权限，并远程登陆。进行扫描端口后发现 waf（192.168.42.6）开启 80 端口（实际生产主机通过防火墙断开与内网除 waf 外所有连接），攻击者发现生产主机存在一个 sql 注入漏洞，想要通过 exp 拿到该网站 webshell。

当攻击者用浏览器打开 192.168.42.6，发现 api.miren.t.n.mi.com 说明 WAF 已经设置成功，如图 1-7。



1- 0

在浏览器测试 get 方式的 sql 注入攻击（非 EXP），

如图 1-8。

http://192.168.42.6/?id=123%20select%20*from%20order%20



1- 0

可以看出非法的访问攻击手段已经被 waf 拦截并记录到 waf 服务器中。查看 waf 日志如图 1-9。



```
root@localhost:/var/log/nginx
File Edit View Search Terminal Help
192.168.42.158 - - [04/Apr/2017:01:45:13 -0700] "GET /favicon.ico HTTP/1.1" 400
42 "http://192.168.42.6/" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
192.168.42.158 - - [04/Apr/2017:01:45:14 -0700] "GET / HTTP/1.1" 400 42 "-" "Moz
illa/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
56.0.2924.87 Safari/537.36"
192.168.42.158 - - [04/Apr/2017:01:45:14 -0700] "GET /favicon.ico HTTP/1.1" 400
42 "http://192.168.42.6/" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
192.168.42.158 - - [04/Apr/2017:01:45:21 -0700] "GET /1.php HTTP/1.1" 400 42 "-"
 "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Ch
rome/56.0.2924.87 Safari/537.36"
192.168.42.158 - - [04/Apr/2017:01:45:21 -0700] "GET /favicon.ico HTTP/1.1" 400
42 "http://192.168.42.6/1.php" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit
/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
192.168.42.158 - - [04/Apr/2017:01:45:23 -0700] "GET /1.php HTTP/1.1" 400 42 "-"
 "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Ch
rome/56.0.2924.87 Safari/537.36"
192.168.42.158 - - [04/Apr/2017:01:45:23 -0700] "GET /favicon.ico HTTP/1.1" 400
42 "http://192.168.42.6/1.php" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit
/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
```

1- 9

由图 1-9 可知，攻击者的攻击手段被记录并拦截下来。并将连接后的数据转移到另一台内网的蜜罐系统中。方便管理员统一收集查看攻击信息。

实验 3，对内网 linux 服务器进行 ssh 爆破（弱口令攻击）

实验环境：centos 6.8 （攻击主机）

Linux 蜜罐系统（kippo 低交互蜜罐开启）

使用一台 linux 对 linux 蜜罐发送 ssh 连接请求，对密码进行猜解。如图 1-10。

```
root@localhost:~/Desktop
File Edit View Search Terminal Help
RSA key fingerprint is 14:d3:8f:4f:26:37:fd:da:76:6e:b9:3f:c5:4b:3c:f1.
Are you sure you want to continue connecting (yes/no)? YES
Warning: Permanently added '192.168.42.63' (RSA) to the list of known hosts.
Password:
Password:
Password:
root@192.168.42.63's password:
123
Permission denied, please try again.
root@192.168.42.63's password:
321Permission denied, please try again.
root@192.168.42.63's password:
Permission denied (keyboard-interactive,password).
[root@localhost Desktop]# ssh 192.168.42.63
Password:
Password:
Password:
root@192.168.42.63's password:
Permission denied, please try again.
root@192.168.42.63's password:
Permission denied, please try again.
root@192.168.42.63's password:
Permission denied (keyboard-interactive,password).
[root@localhost Desktop]#
```

1- 0

这时登陆我们的管理后端查看如图 1-11。

可以看到刚刚的相关操作，登陆的账号和密码信息都有所记录，并且 web 管理后端已经统计成各种图表类型，可以给管理员一个直观的若口弱口令级别，警告管理员密码设置。

Graphical statistics generated from your Kippo honeypot database

Top 10 passwords

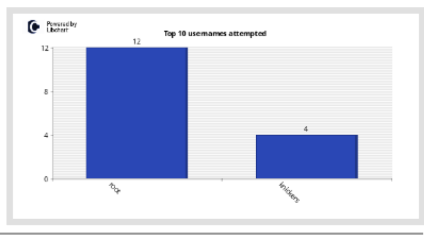
This vertical bar chart displays the top 10 passwords that attackers try when attacking the system.

CSV of all distinct passwords

Top 10 usernames

This vertical bar chart displays the top 10 usernames that attackers try when attacking the system.

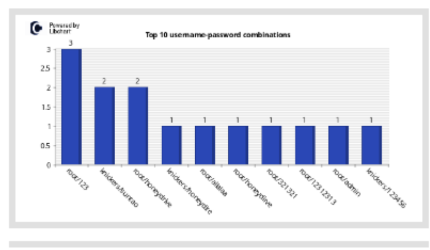
CSV of all distinct Usernames



Top 10 user-pass combos

This vertical bar chart displays the top 10 username and password combinations that attackers try when attacking the system.

CSV of all distinct combinations



1- 0

从图 1-11 可以看出，攻击者使用的密码最多的是什么，爆破用户名次数排行。

This pie chart displays the top 10 username and password combinations that attackers try when attacking the system.

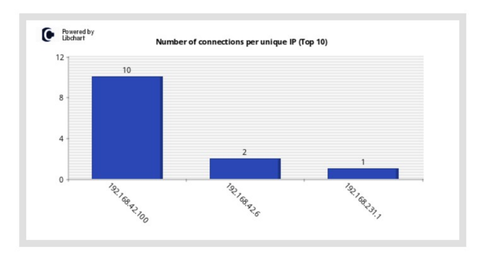
1- 0

如图 1-13、1-14。蜜罐系统已经记录了攻击日期和攻击来源者的 ip，这样对管理员分析内网沦陷机器提供了很大的帮助。也给了以后的日志管理和起诉内容提供了相关的证据。

This line chart displays the weekly activity on the honeypot system. Curves indicate hacking attempts over a weekly period.

CSV of all probes per week

1- 0



1- 0

实验 4、模拟攻击者访问高交互蜜罐系统

当内网一台机器被沦陷后，其他服务器也将处在危险的环境之中，而通过高交互蜜罐系统可以引诱攻击者来攻击这个蜜罐，而本文中的高交互蜜罐当 web 端被攻击者访问时可以记录攻击者 ip 和时间以及页面信息。并将其保存在服务器中。方便管理员查看和分析攻击来源、。

实验环境：windows 主机 （windows 2003server ps2） apache2.0+php7+mysql5

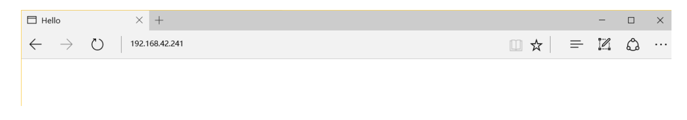
IP：192.168.42.241

Windows 主机（攻击者）

web 客户端

IP：192.168.42.158

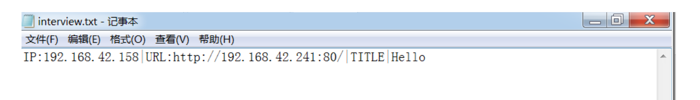
使用 edge 浏览器访问 http;//192.168.42.241/index.php（如图 1-15）



1- 0

假设攻击者通过端口扫描器扫描到了这个服务器开启了 80 服务。并进行访问了。该脚本的作用就是记录访问者 ip 和时间并记录到一个 txt 文件中。

这里我们用管理主机登陆到蜜罐系统的主机查看日志文件。如图 1-16。



1- 0

该蜜罐系统在浏览器设置代理策略，代理到 localhost:8080 下，

这样使得该脚本作用不止如此，假设攻击者已经获得了蜜罐系统的管理员权限后，如果攻击者使用浏览器访问其他服务器所有的流量都将返回到本机，并被后台 php 脚本记录。如图 1-17。

```
IP:192.168.42.158|URL:http://192.168.42.241:80/|TITLE|Hello
IP:127.0.0.1|URL:http://e.firefoxchina.cn:80http://e.firefoxchina.cn/?cachebust=20160321|TITLE|Hello
IP:127.0.0.1|URL:http://e.firefoxchina.cn:80http://e.firefoxchina.cn/?cachebust=20160321|TITLE|Hello
IP:127.0.0.1|URL:http://e.firefoxchina.cn:80http://e.firefoxchina.cn/?cachebust=20160321|TITLE|Hello
IP:127.0.0.1|URL:http://e.firefoxchina.cn:80http://e.firefoxchina.cn/?cachebust=20160321|TITLE|Hello
IP:127.0.0.1|URL:http://baidu.com:80http://baidu.com/|TITLE|Hello
```

1- 0

这样即使攻击者获得管理员权限后也会被记录到所有的访问操作。

5 实验结论

蜜罐技术是继防病毒、防火墙、入侵检测系统等技术过后，又一令人耳目一新的强有力的网络安全技术，本论文将蜜罐系统结合传统防火墙构建的集成蜜罐系统能够大范围覆盖常见内网攻击的手法引诱。由上文试验中可以看出，蜜罐系统对于当前常见的内网攻击手段有着一定的限制作用，来自应用层的攻击本论文模拟的环境可以做到全部网络控制，保证蜜罐系统的数据捕获和与生产系统互不影响。

蜜罐系统的记录功能不止是记录在本机的文件中，也使用了不同的 Database，如 Mysql、PostgreSQL 主要用来储存攻击者的操作记录 IP 地址、时间、ssh 爆破的字典等等。

后端管理采取了多平台多管理的后端，有 php+mysql 和 lua+go 等。统一后端管理都有虚拟化主机管理和镜像备份。

蜜罐容器采取了时下比较成熟和流行的 Vmware 12 虚拟化蜜罐系统操作环境，后期可改进成 docker 或者 QEMU-KVM 等虚拟化。

应用层模拟使用了开源的 kippo 收集和记录 ssh 操作记录，使用 Dionaea+DionaeaFR

组建常见端口模拟和 web 端日志管理程序。Windwos 采用高交互和低交互的模拟模式，通过脚本记录和服务端的日志记录方式反向追踪来源 IP，高交互使用 php+mysql+apache

低交互采取模拟服务端，成功欺骗各类工具的指纹识别。

为了保证管理主机的安全，虚拟环境一定要与主机环境隔离，使用个力度较高的虚拟环境，如 python-virtualenv、QEMU 虚拟器、docker 容器（不推荐）等等，给没他蜜罐系统运行的用户最低的权限，

防止攻击者利用蜜罐服务的漏洞获取蜜罐系统的root权限。

完善的防火墙设置规则，WAF设置规则能够防护cc攻击，sql注入攻击，xss注入攻击，xml注入攻击，http头注入攻击，黑名单设置，并且支持https。生产环境的服务器设置防火墙白名单，使用IPTABLES或者firewall进行设置。Windwos使用相关防火墙设置数据来源白名单。

下一步我们将完善蜜罐系统的通知功能，使用nodejs实时跨平台通知管理员服务器状态，有安全问题做到第一时间通知和记录，保证高交互蜜罐的日志安全性能，进一步提高蜜罐系统的处理效率，完善WAF防火墙。

参考文献：

[1] 毕凯，周炜，基于蜜罐的安全系统设计．计算机工程与设计，2010-11-28．

[2] 蜜罐系统设计的一些想法[J]、网络安全技术与应用

[3] 诸葛建伟，唐勇，韩心慧， 蜜罐技术研究与应用进展[J]. 软件学报，2013,24（4）：825-839